A Unified Model for Collaborative Sentiment Classification

Author Name[1]*
Author Name,
Author Name
[1]address
email address

Abstract

Many existing solutions perform document-level sentiment classification based on a single document only, ignoring other texts that might contribute to better classification accuracy. In this paper, we propose a novel *collaborative sentiment classification* model named CSC. CSC is inspired by the collaborative filtering technique that users with similar rating behaviours in the past are likely to have similar ratings in future. In CSC, we speculate that users with similar rating behaviours are more likely to express similar sentiments toward a product. The proposed CSC model consists of three main components, namely, user-product interaction (UPI) component, document encoding (DE) component, and speculative similar document (SSD) component. The UPI component models user-product interactions, and encodes user/product ratings behaviours into user/product embeddings. The DE component utilizes learned user/product embeddings to capture the informative word vectors for comprising more accurate document representations. The SSD component aggregates documents written by similar user toward the same product for collaborative sentiment classification. Because the user similarities are calculated based on user embeddings that encode user rating behaviours, the aggregated documents are more likely to have similar sentiments. The three components are seamlessly integrated into a unified model. In the unified manner, these three components are jointly optimized, and they mutually complement each other to enhance sentiment classification. We conduct extensive experiments on two public datasets, and demonstrate the advantage of the proposed CSC model over state-of-the-art baselines.

Introduction

Sentiment classification, the task of classifying sentiment or polarity of a given text, has a wide range of applications due to the rapid growth of review data online. This problem can be approached at different levels of granularity (*e.g.,* aspect-, sentence-, and document-levels). In this work, we focus on document-level sentiment classification which is to determine the overall sentiment of a user review about a product.

Various methods have been proposed to tackle sentiment classification, from feature-engineering based methods (Taboada et al. 2011; Li et al. 2014) to deep-learning based models (Tai, Socher, and Manning 2015; Yang et al. 2016; Li et al. 2019b). Recently, several studies (Amplayo et al. 2018; Tang, Qin, and Liu 2015; Chen et al. 2016a) propose to incorporate user and product information into sentiment classification. For example, Chen et al. (2016a) utilize user product embeddings to calculate attention weights over word vectors, and then represent document as a weighted sum of the word vectors. However, existing methods mainly exploit local text information, ignoring sentiment consistency among the documents of the same user/product (Dou 2017). UPDMN (Dou 2017) is the state-of-the-art model that takes advantage of sentiment consistency.[1] The rationale underlying UPDMN is that documents written by (about) a user (product) are more likely to have the same sentiment distribution. Therefore, the model stores documents of each user/product in a memory, and utilizes an attention mechanism to aggregate those documents to produce a document representation for classification. Even though UPDMN has shown promising results, it suffers from the following two problems: (i) for classifying a document, it mainly uses the weighted aggregation of the document representations in the memory while ignoring the document itself; and (ii) documents of a user/product do not necessarily have the same sentiment. Therefore, simply aggregating those documents may introduce noises and negatively affect model performance.

In this paper, we propose a novel **collaborative sentiment classification** (CSC) model. The CSC model is inspired by the basic idea of collaborative filtering, *i.e.,* users with similar rating behaviours in the past are more likely to have similar ratings in the future (He et al. 2017). We therefore speculate that users with similar historical rating behaviours are more likely to write documents expressing similar sentiments toward a product. In short, CSC lever-

*Primarily Mike Hamilton of the Live Oakfile Press, LLC, with help from the AAAI Publications Committee

[1]We note that Amplayo et al. (2018) explore similar users/products to address the cold-start problem, where similar users/products are used to calculate attention weights over the word vectors. In term of text information exploitation, the method limits to the information in a single text.

ages user similarities to select informative documents of the products for collaborative sentiment classification. The selected documents are termed as **speculative similar documents** (SSDs) in this paper. To accomplish this, we first model **user-product interactions**, to collaboratively factorize user/product rating behaviours into low dimensional embeddings. Here, the user embeddings reflect user rating behaviours, and can be used to find similar users. Meanwhile, we encode documents into high-level representations. To capture informative word vectors to represent documents, we incorporate user product embeddings into the **document encoding** process with an attention mechanism. Finally, given a document d written for a product and all the other documents about the same product, we select SSDs written by users similar the user of d. Then we aggregate the representations of those SSDs to collaboratively determine the sentiment of d.

In our model, the modeling of user-product interactions, document encoding, and speculative similar documents are integrated into a unified model. Even though the modeling of user-product interactions and document encodings have been explored earlier (Chen et al. 2016a; He et al. 2017), no existing studies put all components in a unified model. One major advantage of a unified model is that the aforementioned components mutually reinforce each other to enhance sentiment classification. In particular, the modeling of user-product interactions can learn task-specific user product embeddings, which guides the document encoding to capture the most informative word vectors for better document representations. In addition, SSDs are selected based on the similarities between user rating behaviours, which are encoded in user embeddings learned from user-product interactions. To summarize, the contributions of this work are as follows:

- We propose a novel method to select informative documents as auxiliary information for collaborative sentiment classification. The method is based on the speculation that users with similar rating behaviours are more likely to express similar sentiment toward a product.
- We seamlessly incorporate user-product interactions modeling, document encoding, and speculative similar documents modeling into a unified model. These three components are interdependent, and complementary to each other for more accurate sentiment classification.
- We demonstrate that the proposed CSC model outperforms state-of-the-art models on benchmark datasets. We also validate the effectiveness of each of the components of the proposed model.

Related Work

Sentiment classification is a fundamental task in the area of sentiment analysis (Pang, Lee, and Vaithyanathan 2002). The task can be tackled at different levels of granularity, from aspect-level (Li et al. 2019a; Xu, Mao, and Chen 2019; Amplayo and Hwang 2017) to document-level (Amplayo et al. 2018; Dou 2017) depends on the specific application. In this paper, we focus on document-level sentiment classification, which recognizes the overall sentiment of a document. Recent studies (Diao et al. 2014; 2014; Yang et al. 2017) have demonstrated that incorporating additional contexts (*e.g.,* user and product information) significantly improves the classification accuracy. The authors (Tang, Qin, and Liu 2015) utilize user/product text preference matrix to modify the semantic meaning of words. They then apply convolution and average pooling operations to the words for modeling semantic representations of sentences. Chen at al. (2016a) incorporate user product embeddings in modeling the document representations via an attention mechanism. The HCSC model (Amplayo et al. 2018) addresses the cold-start problem in sentiment classification. It leverages similar users/products to obtain additional shared document vectors. These studies mainly utilize local documents, and ignore other documents of the products that might be helpful for sentiment classification.

The UPDMN model (Dou 2017) is the state-of-the-art model that incorporates all documents of users/products for sentiment classification. However, it simply aggregates all documents via an attention mechanism, to comprise the predictive vector. This way inevitably introduces noises and negatively affects the model performance. On the contrary, our proposed model is based on the speculation that users with similar rating behaviours are more likely to express similar sentiments toward a product. Hence our model is able to selectively capture informative documents for collaborative sentiment classification.

The Proposed CSC Model

Before we present our models, we define the following notations. The data records in this work are presented as tuples: $t_k = \langle u_i, p_j, d_k \rangle$, where $d_k \in D$ is a review document, $u_i \in U$ and $p_j \in P$ are the corresponding user and product. In other words, this tuple indicates that user u_i has written a review d_k about product p_j. Our task is to predict the sentiment distribution or rating for tuple t_k.

Each review document is a sequence of words, $d_k = \{w_1^k, \cdots, w_{l_k}^k\}$, where l_k denotes the length of document d_k. For a given product p_j, let $D(p_j)$ denote all the review documents for p_j, and $U(p_j)$ be the corresponding users who write those documents in $D(p_j)$.

CSC Architecture Overview

The architecture of collaborative sentiment classification (CSC) model is presented in Fig. 1. CSC can be decomposed into three components, namely user-product interaction component (UPI), document encoding component (DE), and speculative similar document (SSD) component.

For a given tuple $\langle u_i, p_j, d_k \rangle$, the UPI component takes u_i, p_j as input, and outputs a user-product interaction vector $\mathbf{z}_{ij}$. Specifically, we map the user and product into low dimensional embeddings, and apply multi-layer perceptrons on these embeddings to capture their interactions. The user and product embeddings are randomly initialized and then learned during the training process.

The input of DE component are the words in d_k, and the output is a document vector $\mathbf{d}_k$. In our model, we map the words into word embeddings, and employ stacked 1-

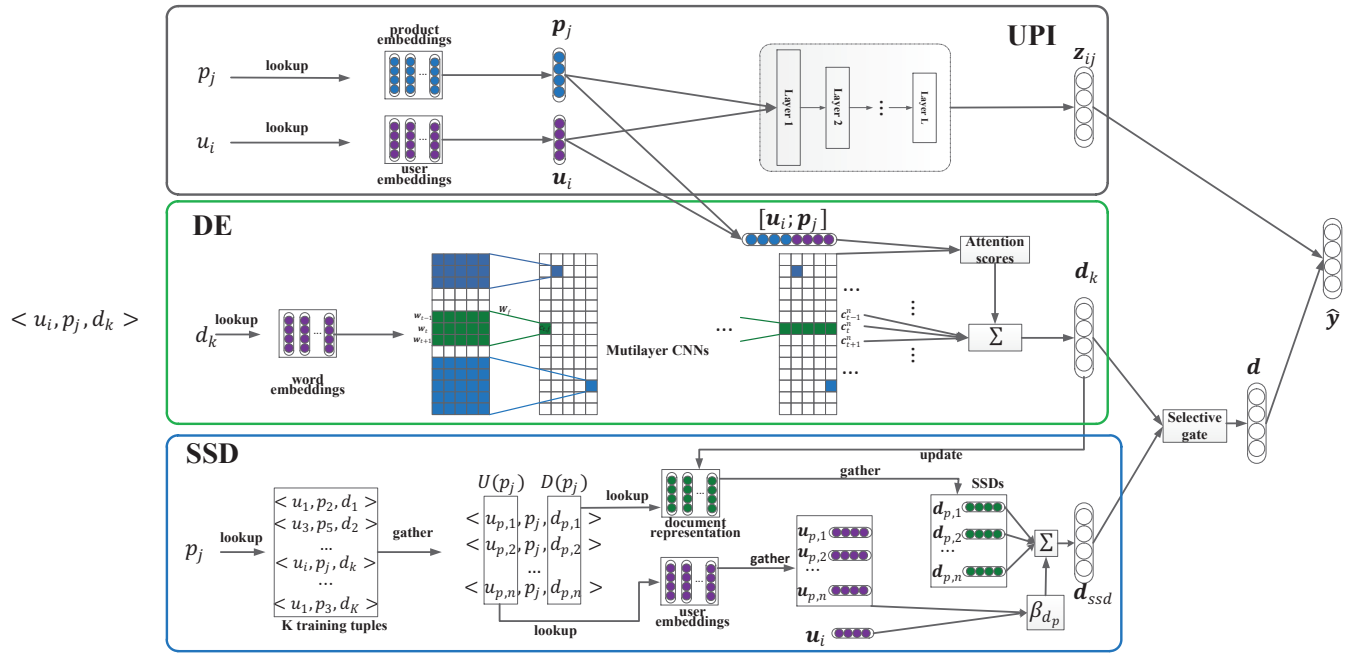


Figure 1: Overall architecture of the collaborative sentiment classification (CSC) model.

dimension Convolution Neural Networks (CNNs) to extract high-level word vectors from those embeddings. We then incorporate user product embeddings (*i.e.,* $\mathbf{u}_i$ and $\mathbf{p}_j$) to capture the most informative word vectors for comprising the document vector $\mathbf{d}_k$.

In SSD component, we select similar users to u_i from $U(p_j)$, and aggregate their corresponding documents from $D(p_j)$ into a SSD vector $\mathbf{d}_{ssd}$. Note that the similarities between u_i and users in $U(p_j)$ are calculated based on their embeddings which encode the user rating behaviours. The selected users are expected to have similar rating behaviours as u_i, hence the corresponding documents are more likely to express similar sentiment as d_k.

User-Product Interaction

The UPI component is based on the latent factor model that decomposes a rating matrix into low dimensional user and product embeddings. The interactions between user-product embeddings capture user preferences over the products; hence they can be used for predicting sentiment ratings. The rationale underlying UPI modeling is that users with similar historical rating behaviours are expected to have similar ratings in future as well.

Inspired by the neural collaborative filtering model (He et al. 2017), we employ multiple neural networks as the basic model to capture deep user-item interactions. We denote $\mathbf{u}_i$ and $\mathbf{p}_j$ as the embeddings of user u_i and product p_j respectively. These embeddings are randomly initialized and learned during training. Given the embeddings, the sentiment class distribution of u_i over p_j is estimated as:

$$\begin{aligned} \mathbf{z}_0 &= [\mathbf{u}_i; \mathbf{p}_j; \mathbf{u}_i \circ \mathbf{p}_j] \\ \mathbf{z}_{ij} &= \phi_L(...\phi_1(\mathbf{z}_0)...) \\ \phi_l(\mathbf{z}_{l-1}) &= \sigma_l(\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l), l \in [1, L] \end{aligned} \quad (1)$$

In the above equation, $[;]$ is the concatenation operation, $\circ$ is the element-wise multiply operation. L is the number of hidden layers in the neural network, and $\mathbf{W}_l, \mathbf{b}_l, \sigma_l$ are the weight matrix, bias vector, and non-linear activation function of the l-th layer, respectively. $\mathbf{z}_{ij}$ can be regarded as the high-level UPI representation. Note that UPI modeling learns task-specific user product embeddings. The learned embeddings are involved in computing attention scores over the words for better document representations.

Document Encoding

The DE component takes a document d_k as input and outputs a distributed document representation $\mathbf{d}_k$. Each word w_t^k is mapped into a low dimensional embedding $\mathbf{w}_t^k \in \mathbb{R}^{d \times 1}$ through an embedding matrix using its vocabulary index. The word embeddings are then input to stacked 1-dimension CNNs to extract high-level word vectors. The CNN layer is applied here because CNNs are shown powerful for sentiment classification (Kim 2014; McCann et al. 2017). They are able to capture local semantics of different levels of granularity when filter maps of different sizes are applied. Specifically, we apply a filter map $\mathbf{W}_f \in \mathbb{R}^{s \times d}$ of size s over a windows of s words centered at word w_t^k, and produce a new single feature $c_{t,f}$:

$$c_{t,f} = g([\mathbf{w}^k_{t-(s-1)/2}; \cdots; \mathbf{w}^k_{t+(s-1)/2}]^T * \mathbf{W}_f + b_f) \quad (2)$$

Here g is a non-linear function, b_f is a bias scalar and $*$ is the convolution operation. In practice, we use filter maps of different sizes (*i.e.,* $\mathbf{W}_{f_1}, \mathbf{W}_{f_2}, \cdots$) to perform the convolution operation, and concatenate the corresponding feature vectors into a vector for word w_t^k:

$$\mathbf{c}_t = [c_{t,f_1}; \cdots; c_{t,f_i}; \cdots; c_{t,f_n}]^T \tag{3}$$

where $\mathbf{c}_t \in \mathbb{R}^{n \times 1}$ can be viewed as the word vector extracted from CNNs for word w_t^k, and c_{t,f_i} is the single feature produced by filter map $\mathbf{W}_{f_i}$ center at w_t^k. Notice that $\mathbf{c}_t$ has dimensionality equals to the number of filter maps. The produced feature vectors $\{\mathbf{c}_t\}_{t=1}^{l_k}$ are then input to multi-layer CNNs to obtain high-level word vectors. We denote the word vectors after n-layer CNNs as $\{\mathbf{c}_t^n\}_{t=1}^{l_k}$.

Words in the documents are noisy. Not all words are equally informative in representing a document, because users may have different preferences over words for describing different products. Therefore, we employ an attention mechanism to capture the informative word vectors, and aggregate them into document representations:

$$\mathbf{d}_k = \sum_{t=1}^{l_k} \alpha_t \mathbf{c}_t^n \tag{4}$$

where $\mathbf{d}_k$ can be viewed as the representation of document d_k. α_t is the attention score assigned to the t-th word in d_k. The attention scores $\{\alpha_t\}_{t=1}^{l_k}$ need to be specialized by the corresponding word, user, and product. They can be obtained as follows,

$$\begin{aligned} \beta_t &= \mathbf{v}^T tanh(\mathbf{W}_c \mathbf{c}_t^n + \mathbf{W}_{up}[\mathbf{u}_i; \mathbf{p}_j] + \mathbf{b}_d) \\ \alpha_t &= \frac{exp(\beta_t)}{\sum_{t=1}^{l_k} exp(\beta_t)} \end{aligned} \tag{5}$$

where the matrices $\mathbf{W}_c, \mathbf{W}_{up}$ and the vectors $\mathbf{b}_d$, $\mathbf{v}$ are model parameters. β_t denotes the relevance score of $\mathbf{c}_t^n$ in representing document d_k and α_t is the softmax normalization of β_t.

Speculative Similar Document

Documents about a user/product reveal the attributes about the user/product from different points of view. Recent studies (Zheng, Noroozi, and Yu 2017) show that all documents of a user/product can be exploited to alleviate data sparsity problem and boost application specific performance (*e.g.,* recommendation). However, in sentiment classification, incorporating reviews of different sentiment ratings has negative impact on the classification performance. To address this issue, the proposed SSD component aggregates only documents written by those users who show similar rating behaviour.

Without loss of generality, given a tuple $t_k = \langle u_i, p_j, d_k \rangle$, we select documents in $D(p_j)$ based on similarities between the rating behaviours of u_i and that of users in $U(p_j)$. Those selected documents are speculated to have similar sentiment distributions, and are aggregated to obtain a SSD representation:

$$\mathbf{d}_{ssd} = \sum_{d_p \in D(p_j)} \beta_{d_p} \mathbf{d}_p \tag{6}$$

where $\mathbf{d}_{ssd}$ is the SSD representation of d_k, $\mathbf{d}_p$ is the document representation of $d_p \in D(p_j)$ obtained with multi-layer CNNs as described in Eq. (4), and β_{d_p} is the corresponding aggregation weight of d_p. The aggregation weights are parameterized by the similarity between u_i and $u_p \in U(p_j)$:

$$\begin{aligned} s(u_i, u_p) &= \mathbf{v}_{ssd}^T tanh(\mathbf{W}_u \mathbf{u}_i + \mathbf{W}_p \mathbf{u}_p + \mathbf{b}) \\ \beta_{d_p} &= \frac{I(u_i, u_p) exp(s(u_i, u_p))}{\sum_{u_p \in U(p_j)} I(u_i, u_p) \cdot exp(s(u_i, u_p))} \\ s.t.\ u_p &\in U(p_j) \end{aligned} \tag{7}$$

where $\mathbf{u}_p$ is the embedding of u_p, $\mathbf{W}_u, \mathbf{W}_p$ are weight matrices and $\mathbf{b}$ is a bias vector, $\mathbf{v}_{ssd}$ is a weight vector. $s(u_i, u_p)$ measures the similarity between u_i and u_p. $I(u_i, u_p)$ is an indication function used to select similar users. In this work, we adopt the thresholding mechanism (Wang et al. 2017; 2016) to define $I(u_i, u_p)$, which equals 1 if the similarity between the two users exceeds a certain threshold θ_s, and 0 otherwise.

$$I(u_i, u_p) = \begin{cases} 1 & s(u_i, u_p) > \theta_s \\ 0 & s(u_i, u_p) \leq \theta_s \end{cases} \tag{8}$$

Notice that θ_s is not a constant in this work, but rather a parameter left for the proposed model to learn (next section). Therefore, the document d_p is not selected for aggregating u_i's SSD representation, if the similarity between its user and u_i is below θ_s. The underlying rationale is straightforward, not all the documents written about a product have the same sentiment, and simply include all the documents for calculating the SSD representation may inevitably introduce noises.

Sentiment Classification

Given the representation of a document d_k, $\mathbf{d}_k$, and the aggregated representation of its SSDs, $\mathbf{d}_{ssd}$, we propose to employ a dual selective gate network to select between $\mathbf{d}_k$ and $\mathbf{d}_{ssd}$ into a pooled document vector, $\mathbf{d}$.

$$\begin{aligned} \mathbf{g} &= \sigma(\mathbf{W}_g[\mathbf{d}_k; \mathbf{d}_{ssd}] + \mathbf{b}_g) \\ f(\theta_s) &= \sigma((t_s - \theta_s)(\theta_s - t_d)) - 0.5 \\ \mathbf{d} &= (1 - f(\theta_s)) * \mathbf{g} \circ \mathbf{d}_k + f(\theta_s) * (1 - \mathbf{g}) \circ \mathbf{d}_{ssd} \end{aligned} \tag{9}$$

where $\circ$ is the element-wise multiplication, and the matrix $\mathbf{W}_g$ and the bias vector $\mathbf{b}_g$ are model parameters, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function to limit the elements of its output in the range of [0,1]. t_s, t_d are the averages of the similarity scores of similar users (*i.e.,* users with similarity score exceeds θ_s) and dissimilar users (*i.e.,* users with similarity score smaller than θ_s) respectively. $f(\theta_s)$ denotes the degree of separation between similar users and dissimilar users. Specifically, if t_s and t_d are close to θ_s, then θ_s will be close to 0, indicating small differences between similar

Statistics	IMDB	Yelp 2013
#users	1310	1631
#products	1635	1635
#train docs	67426	62522
#dev docs	8381	7773
#test docs	9112	8671
#docs/user	64.82	48.42
#docs/product	51.94	48.36
#classes	10	5

Table 1: Statistics of the IMDB and Yelp 2013

users and dissimilar users. In this case, there are high uncertainties in selecting the similar users, and we have low confident in the SSD representation, hence large weight is given to the document representation $\mathbf{d}_k$. On the contrary, $f(\theta_s)$ is close to 0.5 if θ_s provides a large degree of separation between the similar users and dissimilar users, then we have high confidence in the similar users and distribute equal weights to $\mathbf{d}_k$ and $\mathbf{d}_{ssd}$ for sentiment classification. Notice that gate vector $\mathbf{g}$ is automatically learned from the training data, while $f(\theta_s)$ encodes domain knowledge. The hybrid of data-driven and knowledge-driven gate values help to select the most informative features for classifying the document sentiments.

The proposed sentiment classifier then transforms $\mathbf{d}$ and $\mathbf{z}_{ij}$ into a predictive vector with a dimensionality that equals to the number of sentiment classes C.

$$\hat{\mathbf{y}} = softmax(\mathbf{W}_d\mathbf{d} + \mathbf{W}_{upi}\mathbf{z}_{ij} + \mathbf{b}_y)$$
$$loss = -\sum_{<u_i,p_j,d_k>\in\mathcal{D}}\sum_{c=1}^{C} y_c log(\hat{y}_c) \quad (10)$$

where $\mathcal{D}$ is the training set, and $\mathbf{W}_d, \mathbf{W}_{upi}$ are weight matrices and $\mathbf{b}_y$ is a bias vector. $\hat{y}_c$ is the c-th element in predictive vector $\hat{\mathbf{y}}$, indicating the probability that the tuple $\langle u_i, p_j, d_k\rangle$ is classified to class c. We use cross-entropy error between predictive labels $\hat{\mathbf{y}}$ and the ground-truth labels $\{y_c\}_{c=1}^{C}$ as the loss.

Experiment

We conduct experiments on publicly available datasets and compare our model against baselines. To comprehensively evaluate the proposed method, we answer the following research questions:

- **RQ1** Can the proposed method outperform the sate-of-the-art sentiment classification methods?
- **RQ2** Are the proposed UPI and SSD components helpful in improving the performance of sentiment analysis?
- **RQ3** How do the key hyperparameters affect the performance of the proposed model?
- **RQ4** What is the running time of the proposed method compared with the state-of-the-art methods?

Datasets and Evaluation

We conduct experiments on two publicly accessible datasets: IMDB and Yelp 2013. These two datasets were collected by Tang et al. (2015) and divided into train, dev, and test sets with a 8:1:1 ratio. Textual reviews in the datasets are tokenized and sentence-splitted using Stanford CoreNLP (Manning et al. 2014). The statistics of the two datasets are reported in Table 1.

Classification on the two datasets are measured using two metrics: accuracy and root-mean-square error (RMSE). Accuracy measures the overall performance of sentiment classification, and RMSE measures differences between predicted and ground truth rating values.

Implementation

We implement the proposed model with TensorFlow[2], and opensource the code[3]. The dimensionality of word, user, and product vectors are set to 300, and the word embedding matrix is initialized with pre-trained Glove embeddings (Pennington, Socher, and Manning 2014). We use stacked 1-dimension CNNs for extracting word representations. For the first layer CNN, the kernel sizes are set to 3 and 5, each with 128 feature maps. As for the second layer CNN, we use a kernel of size 5 with 300 feature maps. To prevent overfitting, we deploy dropout (Srivastava et al. 2014) layers upon all non-linear transformations with a dropout rate of 0.5. The implemented model is trained via stochastic gradient descent over shuffled mini-batches with a batch size of 128. The defined objective function is optimized using Adam (Kingma and Ba 2015) with an initial learning rate of 0.001. We perform early stopping and fine tune the parameters with the dev set. All of the experiments and training are conducted on a NVIDIA GeForce GTX 1070 graphics card with 8G memory.

Baselines

We compare the proposed model with the following baselines:

- HCSC (Amplayo et al. 2018) models user/product representations from similar users/products to mitigate cold-start problem.
- NSC (Chen et al. 2016a) uses a hierarchical LSTM model and an attention mechanism for encoding documents. The user/product information is considered by the attention mechanism.
- UPDMN (Dou 2017) models document representations with a LSTM encoder and modifies the representations with other documents of the user/product as the memory.
- TUPCNN (Chen et al. 2016b) considers temporal rating behaviours for modeling temporal user/product embeddings, and extends the CNN-based classifier with the embeddings.

Among these baselines, HCSC achieves the state-of-the-art results. Note that there are many other models for document-level sentiment classification, but they are demonstrated to be inferior to the aforementioned baselines in previous studies (Amplayo et al. 2018; Chen et al. 2016a).

[2]https://github.com/tensorflow

[3]URL will be released after paper acceptance

Methods	IMDB		Yelp 2013	
	Acc.	RMSE	Acc.	RMSE
UPDMN	0.465	1.351	0.639	0.662
TUPCNN	0.488	1.451	0.639	0.694
NSC	0.533	1.281	0.650	0.692
HCSC	0.542	1.213	0.657	0.660
CSC	**0.578***	**1.198***	**0.679***	**0.646***

Table 2: Accuracy and RMSE of the proposed models and the baselines. * indicates that the improvements over all the other models are statistically significant for $p < 0.01$.

Therefore, we do not compare with those models to avoid redundancy. Additionally, we also implement three versions of the proposed model to investigate the contributions of each of its components, *i.e.,* (i) the CSC-BASE model that excludes both SSD and UPI components, (ii) the CSC-UPI model that includes UPI component on the basis of CSC-BASE, and (iii) CSC-SSD model that includes SSD component on the basis of CSC-BASE.

Performance Comparison (RQ1)

Table. 2 reports the accuracy and RMSE on both datasets. We have the following observations:

- The proposed model, CSC, achieves the best performance in general. It outperforms the best baseline (*i.e.,* HCSC) by a large margin on both datasets and metrics. This justifies the effectiveness of leveraging informative document of the products for collaborative sentiment classification.
- All the other models yield better results than UPDMN. Even though UPDMN exploits other documents of the users/products for sentiment analysis. Those documents of the users/products do not necessarily have similar sentiments. The inconsistent sentiments of different documents could jeopardise the model and negatively affect its performance. The inferior performance of UPDMN demonstrates the necessity to speculate on the most informative documents for collaborative sentiment classification.
- HCSC and NSC both incorporate user/product embeddings in learning attentive document representations. They achieve significant improvements compared to UPDMN and TUPCNN. This result shows the effectiveness of incorporating user/product preferences for sentiment analysis.
- HCSC and NSC both employ attention mechanism on word vectors to obtain document representations which are then used for sentiment classification. Therefore, both models are limited by the information conveyed in a single document. On the contrary, CSC is able to leverage the informative documents of the products for collaborative sentiment analysis to improve performance.

Efficacy of CSC Components(RQ2)

To investigate the effectiveness of each component (*i.e.,* UPI and SSD) of CSC, we compare CSC with CSC-BASE, CSC-UPI and CSC-SSD in terms of accuracy and RMSE.

Methods	IMDB		Yelp 2013	
	Acc.	RMSE	Acc.	RMSE
CSC-BASE	0.522	1.320	0.633	0.706
CSC-UPI	0.541	1.264	0.640	0.688
CSC-SSD	0.572	1.211	0.668	0.659
CSC	**0.578***	**1.198***	**0.679***	**0.646***

Table 3: Accuracy and RMSE of the proposed models and its variants. * indicates that the improvements over all the other variants are statistically significant for $p < 0.01$.

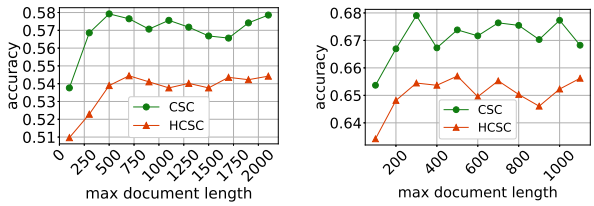
The comparison results are reported in Table 3. We draw the following key observations:

- CSC-UPI yields better classification performance than CSC-BASE on both datasets and metrics. This result demonstrates the effectiveness of incorporating UPI for sentiment analysis. In CSC-UPI, the UPI component is included in the final sentiment prediction, which learns task-specific user/product embeddings. As the document representations are attentively calculated as a function of word, user and product embeddings, the task-specific user/product embeddings help to learn better attentive document representations for sentiment analysis.
- The proposed model achieves better performance than its variants, CSC-UPI and CSC-SSD. The advantage over the variants justifies the efficacy of incorporating the components into a unified model, and demonstrates that the components of the joint model can complement and reinforce each other to enhance sentiment analysis.
- The improvement of CSC-SSD over CSC-BASE is more significant than that of CSC-UPI over CSC-BASE. This shows that SSD component are more effective in improving classification performance.

Evaluation of Impact of Hyperparameter (RQ3)

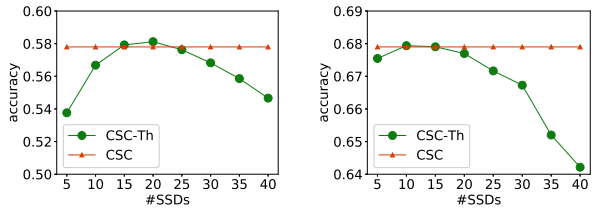
Max document length. To accelerate the training process, documents of various length are padded to a pre-defined maximum length. We investigate the impact of the maximum length on model performance. The classification performance is presented as a function of the maximum document length in Fig. 2. From the figures, we can conclude that the proposed model outperforms the best baseline with respect to different maximum document lengths, justifying its stability and robustness. In Fig. 2, the classification accuracy increases with maximum document length, and retains at a rather stable level afterwards. This indicates that the first few hundred words are sufficient to determine the document sentiments. The inclusion of additional word vectors do not significantly affect the classification performance. As the training time increase along with the maximum document length, we experimentally select maximum document lengths (500 for IMDB and 300 for Yelp 2013) to balance the tradeoff between accuracy and training overhead.

Number of SSDs (#SSDs). In this work, the SSDs are selected based on user similarities for each training tuple, and



(a) IMDB (b) Yelp 2013

Figure 2: Sentiment classification accuracy with respect to different maximum document length.



(a) IMDB (b) Yelp 2013

Figure 3: Sentiment classification accuracy with respect to different number of specular similar documents.

the similar users are automatically identified with the proposed thresholding mechanism. However, to investigate the effectiveness of the thresholding mechanism, we show the performance of a variant (CSC-Th for short) of CSC that selects SSDs based on user similarities in a descending order. Fig. 3 presents the classification accuracy with respect to different #SSDs. For comparison, we also plot the accuracy achieved by CSC, which is supposed to remain constant with respect to different #SSDs.

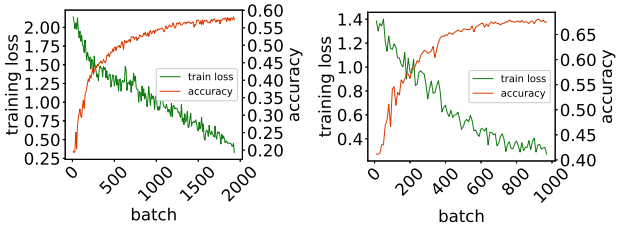
As shown in the figure, we can observe that CSC-Th achieves the optimal performance at some points, however, it is sensitive to #SSDs. In some cases, the performance of CSC-Th even degrades to that of CSC-UPI, where the SSD component does not bring any performance improvement. Moreover, the value of #SSDs for achieving the best performance is different across the datasets, indicating that the optimal #SSDs is rather data-dependent, and it requires a trial-and-error process to tune this parameter. On the contrary, with the combination of the thresholding mechanism and a selective gate, CSC can automatically select appropriate SSDs for each training tuple and yield a stable performance.

Running Time Comparison (RQ4)

In the proposed model, the document representations are calculated by applying 1-dimension CNNs and an attention mechanism on the word vectors. The convolution operations can run in parallel, making it time efficient to train the proposed model. We report the training time in seconds for running 100 batches of data, and present the results in Table 4. For fair comparison, the maximum document lengths in HCSC are set to be the same as those in CSC. Shown in Table 4, HCSC takes 69 and 43 seconds

Models	IMDB	Yelp 2013
HCSC	69.538	43.095
CSC-BASE	13.013 (5.34x)	8.729 (4.94x)
CSC-UPI	13.398 (5.19x)	8.991 (4.79x)
CSC-SSD	13.317 (5.22x)	8.786 (4.91x)
CSC	13.524 (5.14x)	9.061 (4.76x)

Table 4: Training time (in seconds) for running 100 batches data of competing models across the datasets. The number in the parenthesis are the speedup of time compared to HCSC.



(a) IMDB (b) Yelp 2013

Figure 4: Training loss and validation accuracy in each batch on IMDB and Yelp 2013 .

to process 100 batches of data from IMDB and Yelp 2013 respectively, while CSC achieves 5.14x and 4.76x speedup on these two datasets. This is because HCSC uses Bidirectional Long Short Term Memory (BiLSTM) networks for extracting document representations, and they cannot be parallelized. Moreover, comparing CSC with its variants, the differences between their training overhead is marginal. This indicates that incorporating SSD and UPI components yield better results without incurring noticeable overhead.

Fig. 4 presents the training loss and validation accuracy of the proposed model in each batch across the datasets. The proposed model only requires 4 and 2 epochs (that are 2108 and 972 batches) respectively to get the best results on IMDB and Yelp 2013. It justifies the necessity to perform early stopping to avoid overfitting.

Conclusion

In this paper, we propose a novel collaborative sentiment classification model named CSC. This model is based on the speculation that users with similar rating behaviours are more likely to write documents expressing similar sentiment toward a product. We therefore propose to exploit those informative documents for collaborative classification. We conduct experiments on two public datasets, and demonstrate the effectiveness of the proposed model against the state-of-the-art baselines. We also investigate different variants of the proposed model to reveal the rationale underlying the advantage of the different components in the proposed model. We further analyze the impact of hyper-parameters and running time of the proposed model.

References

Amplayo, R. K., and Hwang, S. 2017. Aspect sentiment model for micro reviews. In *2017 IEEE International Conference on Data Mining*, 727–732.

Amplayo, R. K.; Kim, J.; Sung, S.; and Hwang, S. 2018. Cold-start aware user and product attention for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2535–2544.

Chen, H.; Sun, M.; Tu, C.; Lin, Y.; and Liu, Z. 2016a. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1650–1659.

Chen, T.; Xu, R.; He, Y.; Xia, Y.; and Wang, X. 2016b. Learning user and product distributed representations using a sequence model for sentiment analysis. *IEEE Comp. Int. Mag.* 11(3):34–44.

Diao, Q.; Qiu, M.; Wu, C.; Smola, A. J.; Jiang, J.; and Wang, C. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 193–202.

Dou, Z. 2017. Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 521–526.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, 173–182.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1746–1751.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Li, F.; Wang, S.; Liu, S.; and Zhang, M. 2014. SUIT: A supervised user-item based topic model for sentiment analysis. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1636–1642.

Li, X.; Bing, L.; Li, P.; and Lam, W. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, 6714–6721.

Li, Z.; Wei, Y.; Zhang, Y.; Zhang, X.; and Li, X. 2019b. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, 4253–4260.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 55–60.

McCann, B.; Bradbury, J.; Xiong, C.; and Socher, R. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems 30*, 6294–6305.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1):1929–1958.

Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K. D.; and Stede, M. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37(2):267–307.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 1556–1566.

Tang, D.; Qin, B.; and Liu, T. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 1014–1023.

Wang, X.; Lu, W.; Ester, M.; Wang, C.; and Chen, C. 2016. Social recommendation with strong and weak ties. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, 5–14.

Wang, X.; Hoi, S. C. H.; Ester, M.; Bu, J.; and Chen, C. 2017. Learning personalized preference of strong and weak ties for social recommendation. In *Proceedings of the 26th International Conference on World Wide Web*, 1601–1610.

Xu, N.; Mao, W.; and Chen, G. 2019. Multi-interactive memory network for aspect based multimodal sentiment analysis. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, 371–378.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A. J.; and Hovy, E. H. 2016. Hierarchical attention networks for document classification. In *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489.

Yang, M.; Mei, J.; Ji, H.; Zhao, W.; Zhao, Z.; and Chen, X. 2017. Identifying and tracking sentiments and topics from social media texts during natural disasters. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 527–533.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 425–434.